Introduction:
What is modeling?

Modeling a data integration system is defining a correspondence between data tuples at the source and those of the global schema.

e.g.

\{LAV, GLAV, P2P, GAV\}
Introduction:
It is the way you look at it:

– Source-centric: local-as-view or LAV.
– Global-schema-centric: global-as-view or GAV.
– A mixed approach: GLAV.
– Mapping between sources: P2P.

Introduction:
HARD vs EASY

– A HARD problem is a problem that is either hard to analyze and/or hard to compute.

Data Integration Framework:
Problem domain:

Class of data integration systems of our concern:

– Data integration systems that assume one or more data source S, one global mediated schema G and a semantic mapping M that translates S to G.
– Hence data integration system I could be formalized as follows:
    
    I is <G,S,M>
Data Integration Framework:

What is a Mapping?
- A mapping a set of assertions that are used for semantic translation.

What is an Assertion?
- An assertion is a statement in form \( Q_x \rightarrow Q_y \) stating that the concept expressed by \( Q_x \) on schema X is the same concept expressed by \( Q_y \) on schema Y.

Modeling:

Modeling frameworks of our concern:
- LAV
- GAV

What is the difference?
- The rest of the process depends mainly on the approach you choose.
- e.g.
  - The way you define mappings.
  - Integrity constraints?
  - Query processing.

The LAV framework:

Definition:

Restrict the assertions in the mapping to
- All the mappings are from \( Q_s \rightarrow Q_g \)
- Only one element of S is in the Qs part

new form: \( s \rightarrow Q_g \)
The LAV framework:

Advantages:
- when global schema is well established and hard to alter.
  - e.g.
    » in organizations.
    » Ontologies.
  - Because the global schema is an independent factor in the process of defining mappings.
- More extensible: adding a new source does not require changing the mapping scheme.

Limitations:
- No integrity constraints on global schema.
- HARD query processing.

The GAV framework:

Definition:

Restrict the assertions in the mapping to
- All the mappings are from Qg -> Qs
- Only one element of G is in the Qg part

new form: g -> Qs

Advantages:
- Straight forward query processing.
- It allows for enforcing integrity constraints on the global schema.

Limitations:
- Global schema is a dependant factor.
- LAV is more extensible.
  - Adding a new source may entail change in the global schema.
The GAV framework:

The twist:
- Since that global schema is a dependant factor GAV is widely adopted in the web data integration problems.
- Integrity constraints incur additional HARDness to the problem.
  - Inconsistency.
  - Intractability.

Query Processing in LAV:

Two main strategies:
- View-based Query Rewriting.
- View-based Query Solving.

Query Processing in LAV:

View-based Query Rewriting

Informally:
Rewrite all queries submitted to the system using only relations that are in the global schema in the from clause.

Computability:
Decidable problem.

Complexity:
NP-Complete.
- Solution:
  - restrict languages used to define schema and queries.
Query Processing in LAV:

What if such query does not exist?

– Quit the project!

– Second best solution:
  • Maximally contained Query Rewriting.

Query Processing in LAV:

Maximally contained Query Rewriting

Informally:

Rewrite a query $Q$ using only the relations that are in the global schema producing $Q^-$. Such that $Q^-$ best captures $Q$.

Query Processing in LAV:

View-based Query Answering

Informally:

Find the set of tuples $t$ that answers the query $q$ using a set of views $v$.

Formally:

Find the set of tuples that is sufficient to prove $q$ given the extensions of the query $q$. 
Complexity and decidability of the problem depends on two main notions:

- Assumptions:
  - Sound views.
  - Complete views.
  - Exact views.

- Expressive power of languages used to define S and queries posed to G.

**Query Processing in LAV:**

<table>
<thead>
<tr>
<th></th>
<th>CQ</th>
<th>CQ≠</th>
<th>PQ</th>
<th>Datalog</th>
<th>FOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound</td>
<td>CQ</td>
<td>CQ≠</td>
<td>PQ</td>
<td>Datalog</td>
<td>FOL</td>
</tr>
<tr>
<td>CQ</td>
<td>PTIME</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>undec.</td>
</tr>
<tr>
<td>CQ≠</td>
<td>PTIME</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>undec.</td>
</tr>
<tr>
<td>PQ</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>undec.</td>
</tr>
<tr>
<td>Datalog</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>coNP</td>
<td>undec.</td>
</tr>
<tr>
<td>FOL</td>
<td>undec.</td>
<td>undec.</td>
<td>undec.</td>
<td>undec.</td>
<td>undec.</td>
</tr>
</tbody>
</table>

**Query Processing in GAV:**

Very simple:

- Rename elements in the given query.
- Apply query to source.
- Repeat for all sources.
- Union the results.
Query Processing in GAV:

Global schema:
book (PID, Title, Author)
journal (PID, Title, Year)
article (PID, Title, Crosref)

Source schema:
BOOKS(P,T,A)
Journals(P,T,Y,Z)
Articles(P,T,C,X)

Mapping:

book(P,T,A)    {P,T,A|BOOKS(P,T,A)}
article (P,T,C)    {P,T,C|Articles(P,T,C,X)}

Query Processing in GAV:

Query processing scenario:

Query in FOL:
{P,T| book(P,T,A) or journal(P,T,Y) or article(P,T,C)}

Translated query:

{P,T|Books(P,T,A) or Journals(P,T,Y,Z) or Articles(P,T,C,X)}

Conclusions:
– Query processing is straight forward.
– No query reasoning is needed. (NP-Complete)

Inconsistencies Between Sources:

Definition:

Inconsistent set of source: A set of sources is inconsistent if there is no valid database to represent schema G using data in all sources.

Two sources of inconsistencies:
– Mutually inconsistent sources.
– Sources do not satisfy integrity constraints of Global schema. (GAV only)
Inconsistencies Between Sources:

Solutions:

– Data cleaning:
  • Remove/ignore violating tuples.
  • Relax integrity constraints of sources.

– Relax global integrity constraints.

Reasoning on Queries:

Basic query reasoning needs query containment.

– Query containment is NP-Complete
  – Solution
    • Restrict query languages.